WHAT IS CLAIMED IS:

1. A method of synchronizing routing data with another node in a network, comprising:

receiving routing data;

performing a function on at least a portion of the routing data to produce a first digest, where the first digest comprises substantially less data than the routing data;

receiving a second digest from the other node;

comparing the first and second digests to determine whether they are identical to produce first comparison results; and

exchanging a portion of the routing data based on the first comparison results.

2. The method of claim 1, wherein the function comprises at least one of a checksum or a hash.

3. The method of claim 1, wherein the other node performs the function on a corresponding at least a portion of the routing data stored at the other node to produce the second digest.

4. The method of claim 1, wherein the routing data comprises Open Shortest Path First (OSPF) route advertisements.

5. The method of claim 1, further comprising:

receiving multiple third digests from the other node, where the multiple third digests identify multiple sub-portions of the routing data stored at the other node.

6. The method of claim 5, further comprising:

performing the function on corresponding sub-portions of the routing data that is locally stored to produce multiple local digests.

7. The method of claim 6, further comprising:

comparing the multiple local digests with the multiple third digests to produce second comparison results; and

exchanging further portions of the routing data based on the second comparison results.

8. A first node in a network, comprising:

a plurality of interfaces configured to:

receive routing data, and

receive a first digest from a second node in the network; and

processing logic configured to:

perform a function on at least a portion of the routing data to produce a second digest, where the second digest comprises substantially less data than the routing data,

compare the first and second digests to determine whether they are identical to produce first comparison results,

where the plurality of interfaces are further configured to exchange a portion of the routing data based on the first comparison results.

9. A computer-readable medium containing instructions for controlling a processor to perform a method of synchronizing routing data with another node in a network, the method comprising:

receiving routing data;

performing a function on at least a portion of the routing data to produce a first digest, where the first digest comprises substantially less data than the routing data and where the function comprises at least one of a checksum or a hash;

receiving a second digest from the other node;

comparing the first and second digests to determine whether they are identical to produce first comparison results; and

exchanging one or more portions of the routing data based on the first comparison results.

10. A method for designating nodes as one of a master node or a slave node for synchronizing routing data in a network, comprising:

subdividing routing data stored at a first node into multiple portions;

counting the number of multiple portions to produce a first count;

receiving a first message from a second node at the first node, the first message comprising a second count associated with a number of subdivided portions of the second node's routing data;

comparing the first count with the second count to produce first comparison results;

designating the second node as a slave node based on the first comparison results; and

sending a second message to the second node if the second node is designated as a slave node, where the second message comprises a digest associated with the routing data stored at the first node.

11. The method of claim 10, wherein the first message further comprises a digest associated with routing data stored at the second node.

12. The method of claim 10, further comprising:

performing a function to produce the digest, where the digest produced by the function has substantially less data than the routing data stored at the first node.

13. The method of claim 12, wherein the function comprises at least one of a hash or a checksum.

14. The method of claim 10, further comprising:

designating the first node as a master node based on the first comparison results.

15. The method of claim 12, further comprising:

subdividing each of the multiple portions into multiple sub-portions;

performing the function on each of the multiple sub-portions to produce multiple digests.

16. The method of claim 15, further comprising:

sending a third message to the second node, where the third message comprises the multiple digests.

17. A first node in a network, comprising:

a memory;

an interface configured to:

receive routing data,

store the routing data in the memory, and

receive a first message from a second node, the first message comprising a first count associated with a number of subdivided portions of the second node's routing data;

processing logic configured to:

subdivide routing data stored in the memory into multiple portions,

count the number of multiple portions to produce a second count,

compare the second count with the first count to produce first comparison results,

designate the second node as a slave node based on the first comparison comparison results;

wherein the interface is further configured to:

send a second message to the second node if the second node is designated a slave node, wherein the second message comprises a digest associated with the routing data stored in the memory.

18. A method of using database digests to synchronize routing data between a first node and a second node in a network, comprising:

storing first routing data at the first node;

storing second routing data at the second node;

performing, at the first node, a function on a portion of the first routing data, where the function produces a database digest that has substantially less data than the portion of the first routing data; and

sending the database digest to the second node to synchronize the first routing data with the second routing data.

19. The method of claim 18, wherein the function comprises at least one of a hash or a checksum.

20. The method of claim 18, further comprising:

receiving a first acknowledgment message from the first node based on the database digest, where the acknowledgment message indicates whether the second routing data is synchronized with the first routing data.

21. The method of claim 20, further comprising:

subdividing the portion of the first routing data into multiple subportions; and

performing the function on each of the multiple sub-portions to produce multiple database digests.

22. The method of claim 21, further comprising:

sending the multiple database digests to the second node to synchronize the first routing data with the second routing data.

23. The method of claim 22, further comprising:

receiving a second acknowledgment message from the second node based on the multiple database digests, where the second acknowledgment message indicates whether the multiple sub-portions are synchronized with corresponding sub-portions of the second routing data.

24. A system for using database digests to synchronize routing data in a network, comprising:

a first node configured to store first routing data;

a second node configured to:

store second routing data,

perform a function on a portion of the second routing data, where the function produces a database digest that has substantially less data than the portion of the second routing data, and

send the database digest to the first node to synchronize the second routing data with the first routing data.

25. A data structure encoded on a computer-readable medium, comprising:

first data comprising routing data;

second data comprising an identifier for a node in a network;

third data identifying a portion of the routing data; and

fourth data comprising a first digest of the portion of the routing data, where a function is used to produce the digest and where the digest comprises substantially less data than the portion of the routing data.

26. The data structure of claim 25, wherein the function comprises at least one of a hash or a checksum.

27. The data structure of claim 25, further comprising:

fifth data identifying another portion of the routing data; and

sixth data comprising a second digest of the other portion of the routing data, where the function is used to produce the second digest and where the second digest comprises substantially less data than the other portion of the routing data.

28. A system for using database digests to synchronize routing data between a first node and a second node in a network, comprising:

means for storing first routing data at the first node;

means for storing second routing data at the second node;

means for performing, at the first node, a function on one or more portions of the first routing data, where the function produces a database digest that has substantially less data then a respective one of the one or more portions of the first routing data; and

means sending the database digest to the second node to synchronize the first routing data with the second routing data.

29. A method of synchronizing data with another node in a network, comprising:

performing a function on at least a portion of the data to produce a first digest, where the first digest comprises substantially less data than the at least a portion of the data;

receiving a second digest from the other node;

comparing the first and second digests to determine whether they are identical to

produce first comparison results; and

exchanging a portion of the data based on the first comparison results.

30. The method of claim 29, wherein the function comprises at least one of a checksum or a hash.

31. The method of claim 29, wherein the other node performs the function on a corresponding at least a portion of the data stored at the other node to produce the second digest.

32. The method of claim 29, wherein the data comprises Open Shortest Path First (OSPF) route advertisements.

33. The method of claim 29, further comprising:

receiving multiple third digests from the other node, where the multiple third digests identify multiple sub-portions of the data stored at the other node.

34. The method of claim 33, further comprising:

performing the function on corresponding sub-portions of the data that is locally stored to produce multiple local digests.

35. The method of claim 34, further comprising:

comparing the multiple local digests with the multiple third digests to produce second comparison results; and

exchanging further portions of the data based on the second comparison results.

36. A method of using database digests to synchronize data between a first node and a second node in a network, comprising:

storing first data at the first node;

storing second data at the second node;

performing, at the first node, a function on a portion of the first data, where the function produces a database digest that has substantially less data than the portion of the first data; and

sending the database digest to the second node to synchronize the first data with the second data.

37. The method of claim 36, wherein the function comprises at least one of a hash or a checksum.

38. The method of claim 36, further comprising:

receiving a first acknowledgment message from the first node based on the database digest, where the acknowledgment message indicates whether the second data is synchronized with the first data.

39. The method of claim 38, further comprising:

subdividing the portion of the first data into multiple subportions; and

performing the function on each of the multiple sub-portions to produce multiple database digests.

40. The method of claim 39, further comprising:

sending the multiple database digests to the second node to synchronize the first data with the second data.

41. The method of claim 40, further comprising:

receiving a second acknowledgment message from the second node based on the multiple database digests, where the second acknowledgment message indicates whether the multiple sub-portions are synchronized with corresponding sub-portions of the second data.